Copyright © 1979 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 58, No. 3, March 1979
Printed in U.S.A.

Variable Rate Coding of Speech

By J. J. DUBNOWSKI and R. E. CROCHIERE

(Manuscript received July 31, 1978)

In this paper, we examine a number of concepts and issues concerning variable-rate coding of speech. We formulate the problem as a multistate coder (i.e., a coder that can operate at several bit rates) coupled with a time buffer. We first analyze the theoretical aspects of the problem by examining it in the context of a block processing formulation. We then suggest practical methods for implementing a variable rate coder based on a dynamic buffering approach. We also allude to a multiple user configuration of variable-rate coding for TASI *-type applications. A practical example of a variable rate* ADPCM *coder is presented and applied to speech coding. It is shown that by careful design the algorithm can be made to be as robust to channel errors as that of a fixed rate* ADPCM *coder.*

I. INTRODUCTION

In the design of digital speech coders it is often assumed that the coder and channel operate at fixed bit rates. In reality, however, it is known that speech is an intermittent and nonstationary process, and that in many applications the user demand on a communication system is a variable process. In practice, these intermittent properties can be utilized to make the design of a communication system more efficient. For example, the first property, that of an intermittent source, is utilized in communication systems such as TASI (Time Assignment Speech Interpolation).[1-3] The second property, that of a variable demand on the system, is being explored by various authors for use in packet transmission systems[4,5] and results in a variable rate channel from the point of view of the user.

In both of the above systems, an important element of the system is a variable-rate coder. In its simplest form, it may amount to a trivial transmit/no transmit decision as was used in the initial TASI systems. More generally, we might characterize a variable-rate coder according to a configuration shown in Fig. 1 where both the source activity and the channel rate are assumed to be variable.

In this paper, we examine a number of concepts of variable rate coding. We formulate the problem as a multi-state coder (i.e., a coder with several transmission states) coupled with a buffer to take up the "slack" between the desired source rate and the channel rate. In Section II we investigate theoretical aspects of variable rate coding using block processing concepts and rate distortion theory. Section III covers practical aspects of implementing variable-rate coders and in Section IV we present an example of a variable-rate ADPCM coder.

II. A BLOCK PROCESSING ANALYSIS OF VARIABLE RATE CODING

2.1 Thaoratical consideration

To examine the theoretical performance of a variable-rate versus a fixed-rate coder, we can consider the problem in terms of a block processing problem. Figure 2a illustrates an example of a block of N samples of a zero mean nonstationary signal $s(n)$ as a function of time n. For convenience, we assume that this signal is uncorrelated from sample to sample (as in the difference signal of a DPCM coder).

Figure 2b illustrates the "short-time" variance or power of this signal, denoted as $\sigma^2(n)$. The noise power introduced by the coder is

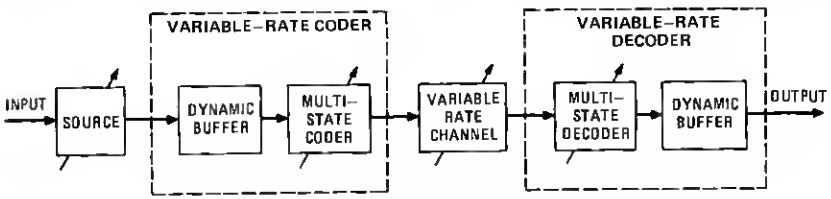

Fig. 1—A general characterization of a variable-rate coder.

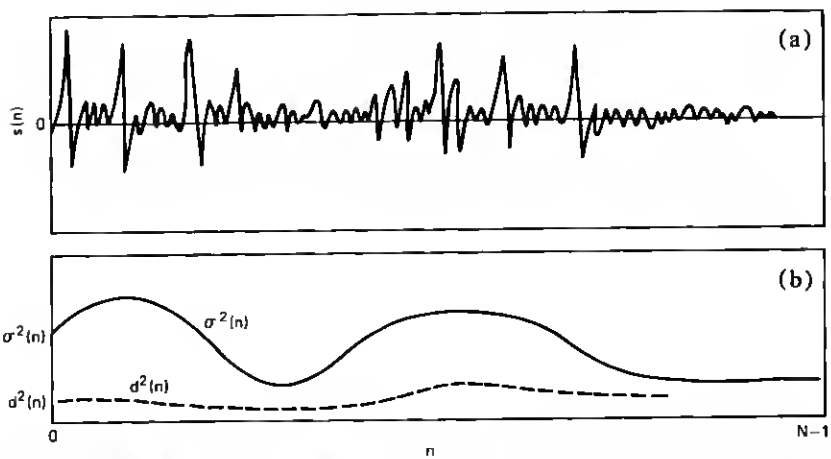

Fig. 2—Illustration of (a) a speech waveform and (b) its variance and quantization distortion after coding.

denoted as $d^2(n)$ and is also illustrated in Fig. 2b. As a performance criterion, we assume that the signal power to noise power ratio over the block, defined as

$$\text{s/n} = 10 \log \left[\frac{\sum_{n=0}^{N-1} \sigma^2(n)}{\sum_{n=0}^{N-1} d^2(n)} \right], \tag{1}$$

is a sufficient measure for comparison. We discuss the practical merits of this measure later.

For a fixed-rate coder, the same number of bits/sample, R_f, is used for quantizing each sample $s(n)$. Therefore, the number of bits, B, used to encode the total block is

$$B = R_f N. \tag{2}$$

Also, from rate-distortion theory,[6,7] it is known that an approximate relationship between the bit rate and distortion of a quantizer is

$$R_f = \theta + \frac{1}{2} \log_2 \left(\frac{\sigma^2(n)}{d^2(n)} \right), \tag{3}$$

where $\sigma^2(n)$ is the variance of the signal as a function of time n, $d^2(n)$ is the variance of the quantization noise, and θ is a constant which is dependent on the characteristic of the quantizer and on the probability distribution of the signal. By rearranging (3), the distortion of the quantizer as a function of time can be shown to be

$$d^2(n) = \sigma^2(n) 2^{2(\theta - R_f)}. \tag{4}$$

By averaging $d^2(n)$ over N samples and applying the results to (1) and (2), the s/n of the fixed rate coder can then be shown to have the form

$$\text{s/n}\Big|_{\substack{\text{fixed} \\ \text{rate}}} = 20 \, (R_f - \theta) \log_{10} 2 \tag{5a}$$

$$= 20 \left(\frac{B}{N} - \theta \right) \log_{10} 2. \tag{5b}$$

This s/n does not include the additional prediction gain that can be obtained if the input to the coder is correlated. For our purposes in this section, we assume that all correlations in the signal have been removed prior to quantization and that this s/n represents only the signal-to-noise ratio of the residual (uncorrelated) signal.

For the variable-rate coder, the number of bits/sample used to encode the nth sample is denoted as $R(n)$ (where it is assumed that $R(n)$ does not have to be an integer). The choice of $R(n)$ for $n = 0, 1, \cdots, N-1$ is then made such that the signal-to-noise ratio in (1) is

maximized and the total number of bits used to encode the block is B, i.e.,

$$B = \sum_{n=0}^{N-1} R(n). \tag{6}$$

The solution to this maximization problem is well known[7,8] and results in the condition that the distortion power $d^2(n)$ at each sample must be identical, i.e.,

$$d^2(1) = d^2(2) = \cdots d^2(N-1) = d_v^2. \tag{7}$$

Therefore, to maximize the block s/n the noise generated by the variable-rate coder must be flat across time. The number of bits/sample which must be used by the coder as a function of time is then

$$R(n) = \theta + \frac{1}{2}\log_2\left(\frac{\sigma^2(n)}{d_v^2}\right). \tag{8}$$

By applying (8) to (6), a relationship between the total number of bits in the block, B, and the distortion d_v^2 can be expressed in the form

$$\begin{aligned} B &= \sum_{n=0}^{N-1} R(n) \\ &= N\theta + \frac{1}{2}\log_2 \prod_{n=0}^{N-1} \sigma^2(n) \\ &\quad - \frac{N}{2}\log_2 d_v^2. \end{aligned} \tag{9}$$

Rearranging terms and solving for d_v^2 gives

$$d_v^2 = 2^{2(\theta - B/N)}\left[\prod_{n=0}^{N-1} \sigma^2(n)\right]^{1/N}. \tag{10}$$

The signal-to-noise ratio for the variable-rate coder can now be determined as

$$\text{s/n}\,|_{\substack{\text{var.}\\ \text{rate}}} = 10\log_{10}\left(\frac{\sum_{n=0}^{N-1}\sigma^2(n)}{Nd_v^2}\right) \tag{11}$$

and substituting in d_v^2 from (10) gives the revealing form

$$\begin{aligned} \text{s/n}\,|_{\substack{\text{var.}\\ \text{rate}}} &= 20\left(\frac{B}{N} - \theta\right)\log_{10} 2 \\ &\quad + 10\log_{10}\left[\frac{\frac{1}{N}\sum_{n=0}^{N-1}\sigma^2(n)}{\left(\prod_{n=0}^{N-1}\sigma^2(n)\right)^{1/N}}\right]. \end{aligned} \tag{12}$$

In comparing the s/n of the variable rate coder (12) to that of the fixed rate coder (5b), it is seen that the first term in (12) is identical to that of the fixed-rate coder. The second term therefore represents the improvement in block s/n that can be expected by using a variable-rate coder instead of a fixed-rate coder. As seen by the form of this term, this improvement is signal-dependent and is in fact equal to the ratio of the arithmetic to geometric means of the signal variance $\sigma^2(n)$ over the block. If $\sigma^2(n)$ varies widely over the block, i.e., if the signal is highly nonstationary, then this gain can be large. If $\sigma^2(n)$ is relatively constant over the block, i.e., the signal is approximately stationary, then the arithmetic and geometric means are essentially equal, and no improvement can be expected.

It is interesting to note that this result is similar in form to that in transform coding.[8] In transform coding, the variation of the signal variance across the block corresponds to a variation in the frequency domain and occurs due to correlations in the input signal (i.e., a nonflatness of the signal spectrum). In the variable-rate coding application, we assumed that these correlations have already been removed and that the variation of the signal variance across the block occurs due to the nonstationarity of the signal in time. By a careful reformulation, however, both the effects of correlated inputs (i.e. prediction gain) and nonstationarity can be incorporated into the above relations for the variable rate coder.

The reader should also be cautioned that while block s/n is appealing mathematically it may not be the most appropriate criterion in terms of perception.[9,10] In practice, some modification of the block s/n criterion may be required. Since little is presently understood about the perceptual effects of the distribution of s/n in time, this is a subject that requires further study before a more perceptually meaningful criterion can be proposed. Further comments on this subject are presented in Section V.

2.2 Potential improvements of variable rate over fixed rate coding

2.2.1 Single speaker

To obtain estimates of the theoretical improvement in block s/n for a variable rate coder, we have measured the arithmetic-to-geometric mean ratio of the signal variance $\sigma^2(n)$ over blocks for speech data for a (one-sided) telephone conversation. The signal variance $\sigma^2(n)$ was obtained by running a 4-bit ADPCM coder[11] on the sentence and using the step-size of the coder as a (scaled) estimate of $\sigma(n)$ of the differential input to the quantizer. Figure 3a shows an example of the speech waveform and Fig. 3b shows the corresponding scaled estimate of $\sigma(n)$ for the differential input to the quantizer in this coder.

The variance estimate $\sigma^2(n)$ of the coder was partitioned into blocks of size N and the ratio

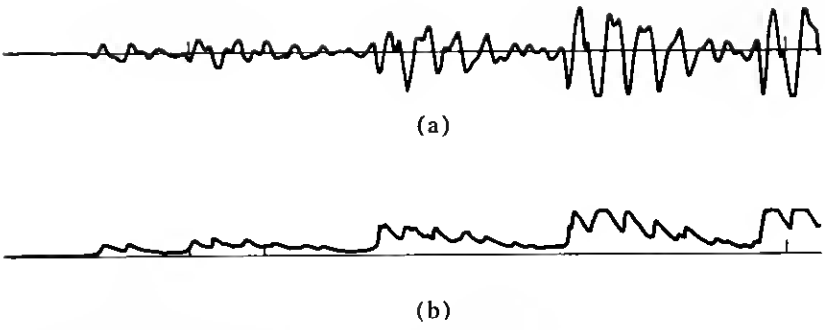

Fig. 3—Example of (a) a speech signal and (b) the estimate of $\sigma(n)$ of its first-order predicted difference signal (8-kHz sampling rate).

$$G = 10 \log_{10} \left[\frac{\frac{1}{N} \sum_{n=0}^{N-1} \sigma^2(n)}{\left(\prod_{n=0}^{N-1} \sigma^2(n) \right)^{1/N}} \right] \text{(dB)} \tag{13}$$

was computed for each block to obtain an estimate of the potential s/n gain for variable rate coding.

The solid line in Fig. 4 shows a plot of the average G, denoted as $\overline{G}$, for this sentence as a function of the block size N in milliseconds. As seen in Fig. 4, significant gains in s/n cannot be expected with variable-rate coding of a single speech source until block sizes greater than about 100 ms are used. That is, the size of the block must be greater than the typical duration of phonemes and micro-silence in speech before improvements in s/n can be realized.

In real-time communications systems, blocks of this size may not be acceptable because they imply large transmission delays. Other potential applications exist, however, in voice-storage and message "store-and-forward" systems where delays may not be of concern. An alternate advantage that is offered with variable-rate coding is that it allows greater flexibility in gracefully varying the transmission rate of the coder rather than restricting it to rates which are a multiple of the sampling rate. Block sizes can be relatively small to achieve this purpose.

2.2.2 Multiple speakers (TASI)

When several sources share a channel, possibilities exist for greater improvements in overall performance due to TASI advantages. One possible approach to encoding P sources into a single channel, in a block fashion, is to assign each source a sub-block of size N/P. By concatenating the P sub-blocks into a single large block of size N, the problem can again be treated as a single source problem.

Figure 5 illustrates such an example where the variances of P concatenated sources $\sigma_1^2(n), \cdots, \sigma_P^2(n)$ are plotted as a function of time n. If the sources have greatly different variances, as depicted in Fig. 5, then the effective concatenated signal over the block will appear

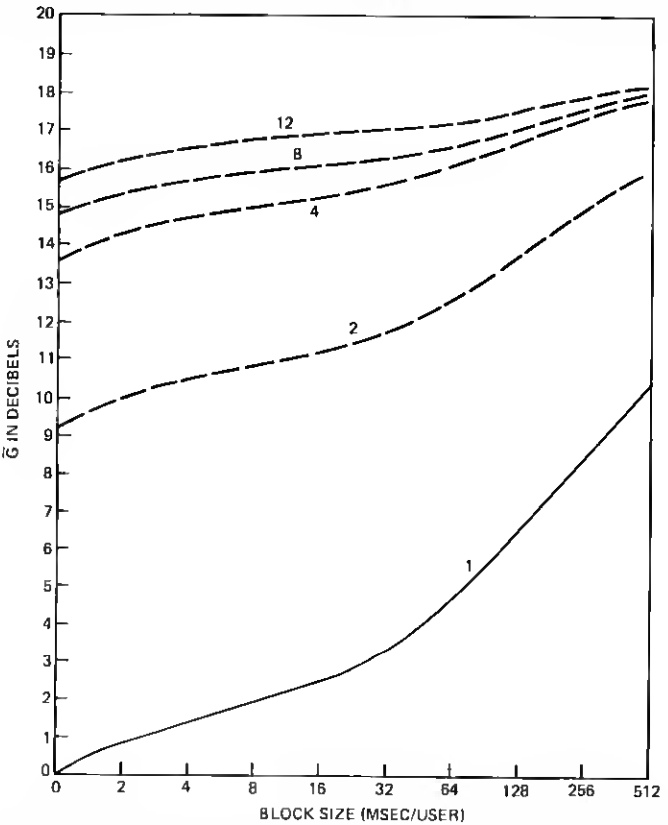

Fig. 4—Arithmetic-to-geometric mean ratio of the signal power (expressed in decibels) of a sentence as a function of the block size.

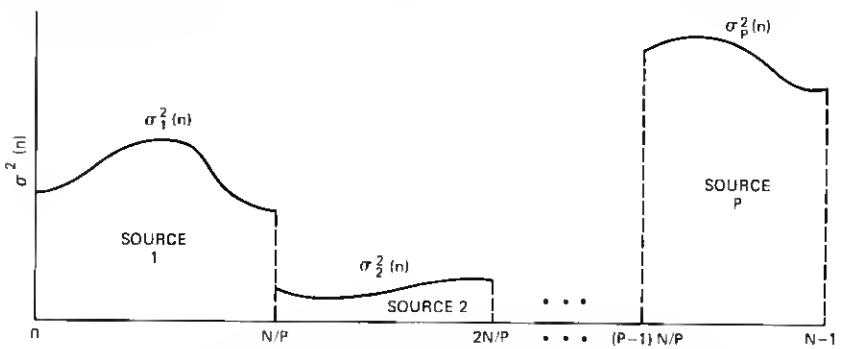

Fig. 5—Block formulation of a multi-user variable rate coder.

to be highly nonstationary and will therefore have a large arithmetic-to-geometric mean ratio. This suggests that a large s/n gain over the block can be obtained using variable-rate coding instead of a fixed bit/sample assignment. In effect, sources with larger variances will receive more bits and sources with lower variances will receive fewer bits. Each source will effectively receive the same amount of noise power as shown by eq. (7). Whether this is the most appropriate choice from a subjective point of view is again a question which remains unanswered at this time.

The dashed lines in Fig. 4 indicate measured values of $\overline{G}$ for 2, 4, 8, and 12 shared users. The gains along the left vertical axis are strictly due to TASI gains alone.

III. PRACTICAL CONSIDERATIONS IN IMPLEMENTING VARIABLE RATE CODERS

3.1 A block processing approach

In Section II, we assumed for purposes of analysis that the variable rate coder is implemented in a block processing manner with a fixed total number of bits B allowed in each block. Practical bit allocation schemes for this type of implementation have been investigated for use in transform coding[7,8] and can be carried over to the variable-rate coding application as well. Since this can be done in a relatively straightforward manner, we will not go into detail on this approach.

3.2 Dynamic buffer approach

An alternative approach to variable rate coding can be realized using a dynamic buffering strategy. A similar approach has been investigated by Tescher and Cox for use in image coding.[12] The method is illustrated in Fig. 6 for a single source example. The coder receives speech samples $s(n)$ at a fixed sampling rate and encodes them with a variable number of bits/sample. The output bits of the coder are then stored serially in a dynamic first-in, first-out buffer and the channel receives output bits from the buffer at the channel rate. A buffer control monitors the state of the buffer and a variance estimate of the input signal and regulates

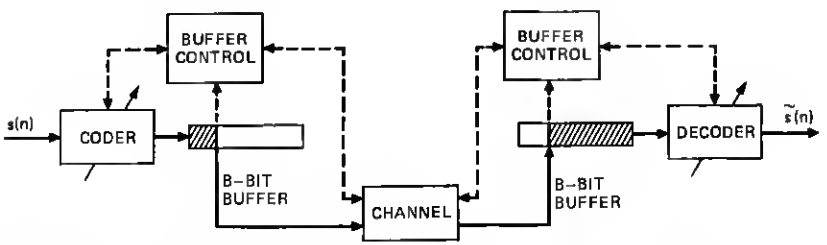

Fig. 6—Block diagram of a variable-rate coder based on buffering the output bit stream.

the number of bits/sample used by the coder. At the receiver, a similar variable rate decoding process takes place.

When the activity in the source is high, the buffer control increases the number of bits/sample used by the coder and decoder respectively above the channel rate. The transmitter buffer begins to fill up, and the receiver buffer begins to drain out. When the source activity is low, the coder and decoder use less than the average number of bits/sample and the reverse process takes place. The total signal delay across the coder/decoder is fixed at a value equivalent to the buffer size, B bits, divided by the channel rate (bits/second).

An alternative dynamic buffer strategy, based on buffering the data samples, is shown in Fig. 7. In this case, the buffer supplies samples to the coder at a variable rate. The buffer control adjusts this rate as a function of buffer status and signal variance while matching the output rate of the coder to that of the channel rate (bits/sample). When the source activity is high, the actual sampling rate transmitted through the channel lags the source rate. This causes the transmit buffer to fill and the receive buffer to empty, as in the scheme of Fig. 5. Conversely, when the source activity is low, the channel transmits samples at a rate greater than the input source rate. This results in filling the receiver buffer while depleting the transmitter buffer. The total signal delay for the system is equal to the buffer size N (samples).

3.3 Buffer control

In both the above dynamic buffering approaches, a key element in the algorithm is the buffer control. In this section, we propose a technique for implementing this control in a recursive manner which applies to either of the above methods.

The algorithm is based on the rate distortion relation (8), which can be expressed in the form

$$R(n) = \log_2 \frac{\sigma^2(n)}{d_s^2(n)}, \tag{14}$$

where $d_s^2(n)$ denotes the (scaled) distortion level in the quantizer and includes the factor θ in (8). Therefore,

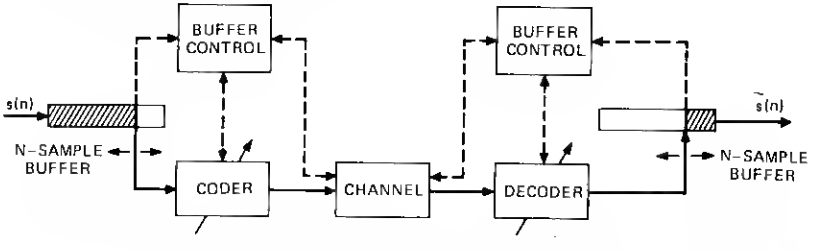

Fig. 7—Block diagram of a variable-rate coder based on buffering the input samples.

$$d_s^2(n) = d_v^2/2^\theta, \tag{15}$$

where $d_s^2(n)$ will be allowed to vary "slowly" with time in a manner which will be described shortly.

In practice, (14) must be modified to account for overflow and underflow of the buffers. Let B denote the size of the buffer and $b(n)$ denote the number of bits stored in the transmitter buffer at time n. Furthermore, let $R_c(n)$ denote the actual number of bits/sample used by the coder at time n and $R_d(n)$ be the number of bits removed from the buffer during the sample period at time n for transmission over the channel. If the transmitter buffer is full, then the coder cannot be permitted to use more than $R_d(n)$ bits/sample and if the buffer is empty the coder cannot be permitted to use less than $R_d(n)$ bits/sample. Therefore, the following algorithm applies:

$$R_c(n) = \begin{cases} [R(n)] \leqslant R_d(n) & \text{if} \quad b(n) = B \\ [R(n)] & \text{otherwise} \\ [R(n)] \geqslant R_d(n) & \text{if} \quad b(n) = 0, \end{cases} \tag{16}$$

where $[R(n)]$ implies the operation of rounding $R(n)$ in (14) to the nearest integer, $\leqslant$ implies reducing $R_c(n)$ to be less than or equal to $R_d(n)$, and $\geqslant$ implies increasing $R_c(n)$ to be greater or equal to $R_d(n)$.

While the constraints in (16) prevent the buffers from overflowing or underflowing, they are not sufficient to assure that the buffers will be effectively utilized. If the average $R(n)$ is too large or too small, the buffers will remain in a state of being near full or near empty, respectively. To efficiently utilize the buffers, the average rate of $R(n)$ should be close to that of the channel rate. This is similar to eq. (6) in the block processing approach, which states that the average bit rate over the block is equal to the channel rate. This condition must be realized by adjusting the distortion level $d_s^2(n)$. If the transmitter buffer is excessively full for long periods of time, then it can be seen that $d_s^2(n)$ is too small and should be increased. Alternatively, if the transmitter buffer is empty for long periods of time, then $d_s^2(n)$ is too large and should be reduced.

The algorithm that we have investigated for controlling $d_s^2(n)$ is based on the recursive relation

$$d_s^2(n) = d_s^2(n-1) \cdot H(b(n-1)), \tag{17}$$

where $d_s^2(n)$ is the distortion level at time n, $d_s^2(n-1)$ is the distortion level at time $n-1$, and $H(b(n-1))$ is a multiplication factor which is dependent on the number of bits $b(n-1)$ in the transmitter buffer at time $n-1$. Figure 8 illustrates an example of $H(b(n-1))$ as a function of $b(n-1)$. The exact shape of $H(b(n-1))$ is not overly critical, except that it should be monotonically increasing and be less than 1 for $b(n-1)$ near zero and greater than 1 for $b(n-1)$ near B. The intercept where $H(b(n-1)) = 1$ determines the average buffer level

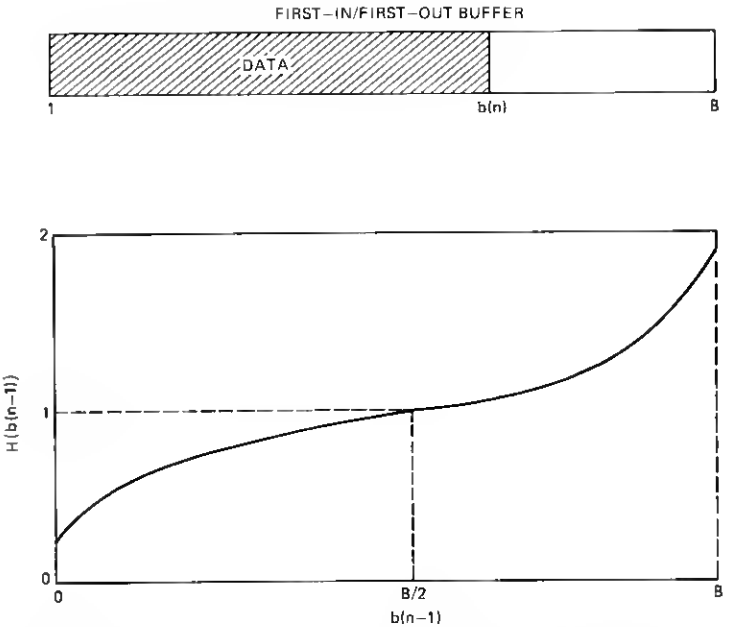

Fig. 8—Multiplier value $H(b(n-1))$ as a function of the buffer status $b(n-1)$.

about which the $b(n)$ fluctuates. If $H(b(n-1))$ is close to 1 for all $b(n-1)$, the algorithm will adapt slowly and if it becomes greatly different than 1, the algorithm will adapt rapidly. Typically, the time constant for adaption should be on the order of the total buffer delay, B. In practice, a piecewise approximation to $H(b(n-1))$ is probably sufficient. Also, it is desirable in practice to set maximum and minimum levels for $d_s^2(n)$, i.e.,

$$d_{\min}^2 \leqslant d_s^2(n) \leqslant d_{\max}^2. \tag{18}$$

This algorithm for controlling $d_s^2(n)$ is similar in many respects to the one-word memory algorithm proposed by Jayant, Flanagan, and Cummiskey[11] for adapting the step-size of an ADPCM coder.

A choice exists in generating the buffer control algorithm at both the receiver and transmitter, or at the transmitter alone. If the latter choice is made, additional information must be transmitted along with the serial data to indicate code word size. In either case, recovery from channel errors is essential. One example for accomplishing this recovery is discussed in the next section.

IV. AN EXAMPLE OF A VARIABLE-RATE ADPCM CODER

4.1 Basic design

To investigate the properties of a variable-rate coder, we have implemented a modified version of the algorithm in Fig. 7 by computer

simulation. A block diagram of this implementation is shown in Fig. 9. The variable rate coder was designed around an ADPCM (adaptive differential PCM) coder[11] that can operate at 2, 3, 4, or 5 bits/sample.

The output of the ADPCM coder is framed into packets of typically 60 bits with a 2-bit header preceding each packet. The buffer control updates the number of bits/sample, $R_c(n)$, used by the ADPCM coder once per packet and transmits this decision to the receiver by means of the 2-bit header. It receives information on the buffer status $b(n)$ from the input buffer and an estimate of the signal variance $\sigma^2(n)$ from the ADPCM coder.

Each packet is encoded with either 2, 3, 4, or 5 bits/sample corresponding to 30, 20, 15, or 12 samples of data per packet respectively. A packet length of 60 bits (plus 2 header bits) is chosen because it is the smallest common multiple of 2, 3, 4, and 5 and results in a fixed packet size independent of the number of bits/sample used by the ADPCM coder.

Because the buffer control transmits the number of bits/sample, $R_c(n)$, used for encoding each packet the receiver algorithm is simplified and does not require a buffer control computation. This, coupled with the fixed packet size, allows for an overall variable rate coder that is more robust in recovering from channel errors. If a buffer control computation were to be used in the receiver, its variance information $\sigma^2(n)$ would have to be obtained from the ADPCM decoder and it would be highly susceptible to channel errors in the data. Once synchronization between the transmitter and receiver is lost it may not be able to recover because the receiver may be using incorrect bits for decoding. With the algorithm proposed here, this cannot happen. Synchronization is unaffected by errors in the data stream. If an error occurs in the header bits, a misalignment of the transmitter and receiver buffers can occur. This type of error is not disastrous, however, and is recoverable with this algorithm as will be seen later.

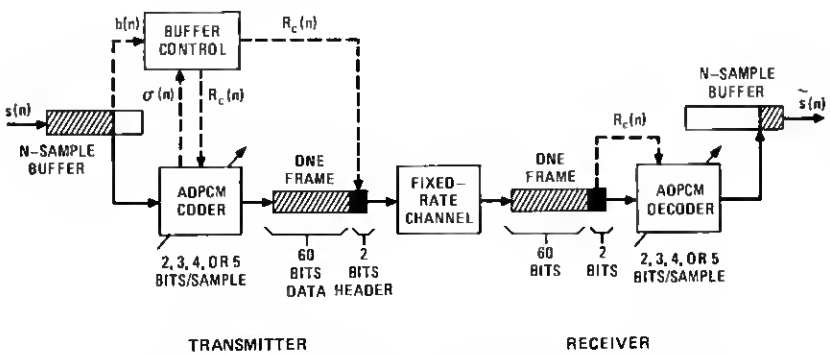

Fig. 9—Block diagram of the variable rate ADPCM coder.

4.2 The ADPCM coder

Figure 10 is a block diagram of the ADPCM coder/decoder. The signal, $e(n)$, resulting from the difference of the input $x(n)$ and its predicted value $y(n)$, is quantized using an adaptive step-size quantizer. The predicted signal, $y(n)$, is obtained from a first-order predictor, as seen in the figure. In the receiver, the difference signal $\hat{e}'(n)$ is decoded from an adaptive step-size decoder and the first-order predictor loop is used to generate the output signal $\hat{x}'(n)$.

The step-size logic adapts the quantizer step-size to track the rms level $\sigma(n)$ of the error signal $e(n)$ and is based on the one-word memory algorithm proposed by Jayant, Flanagan, and Cummisky.[11] Letting $\Delta(n)$ represent the step-size at time n and $\Delta(n-1)$ represent the step-size at time $n-1$, this algorithm is described by the relation

$$\Delta(n) = \Delta(n-1) \cdot M(|\, c(n-1)|)\,. \tag{19}$$

$M(|\, c(n-1)|)$ is a multiplication factor that depends on the magnitude of the code word $c(n-1)$ at time $n-1$. If upper quantizer levels are used, a value of M greater than one is used and if lower quantizer levels are used, a value of M less than one is used. The M values that were used are close to the values proposed by Jayant.[11]

In the operation of the variable rate coder, the number of bits/sample used by the ADPCM coder changes and the step-size must be adjusted accordingly. This adjustment is made in such a way that the

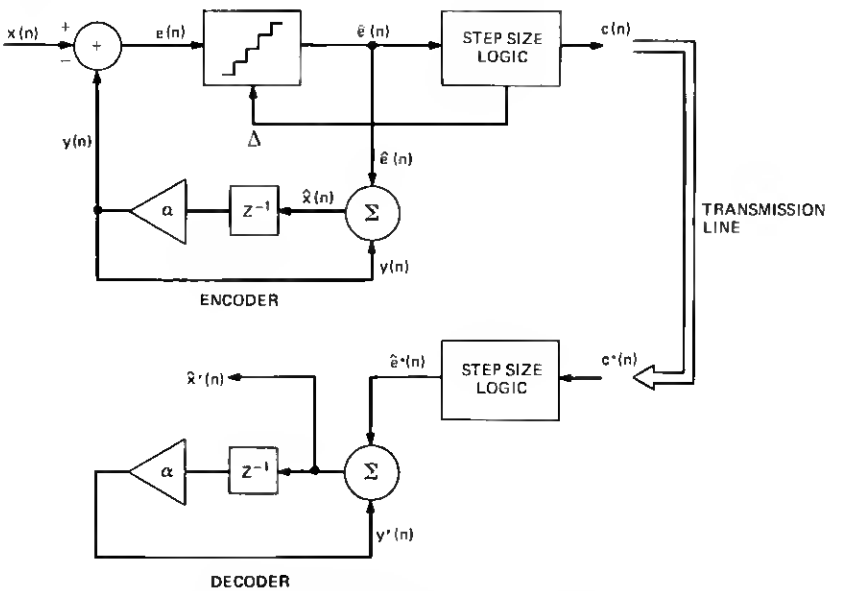

Fig. 10—Block diagram of the ADPCM coder.

center of the quantizer characteristic for the new bit rate is matched to the center for the previous bit rate. This alignment is illustrated in Fig. 11 for the 2-, 3-, 4-, and 5-bit quantizer characteristics. The horizontal scale denotes the (appropriately normalized) input signal $e(n)$ to the quantizer and the vertical scale denotes the (appropriately normalized) output signal $\hat{e}(n)$ from the quantizer (plotted only for positive values of $e(n)$ and $\hat{e}(n)$). The step-sizes Δ_2 to Δ_5 denote the relative step-sizes for the 2- to 5-bit/sample quantizer characteristics, respectively. By adjusting the step-size in this way, the loading factor and the dynamic range of the quantizer remains approximately the same when the number of bits/sample is changed—only the resolution changes.

4.3 Variance estimation

The buffer control requires an estimate of the variance of the signal $e(n)$ to compute the bit allocation $R_c(n)$. This estimate is obtained directly from the step-size adaptation algorithm in the ADPCM coder. It can be observed that, for a given loading factor and a given number of bits/sample, the square root of the signal variance, $\sigma(n)$, of the signal $e(n)$ is proportional to the step-size.

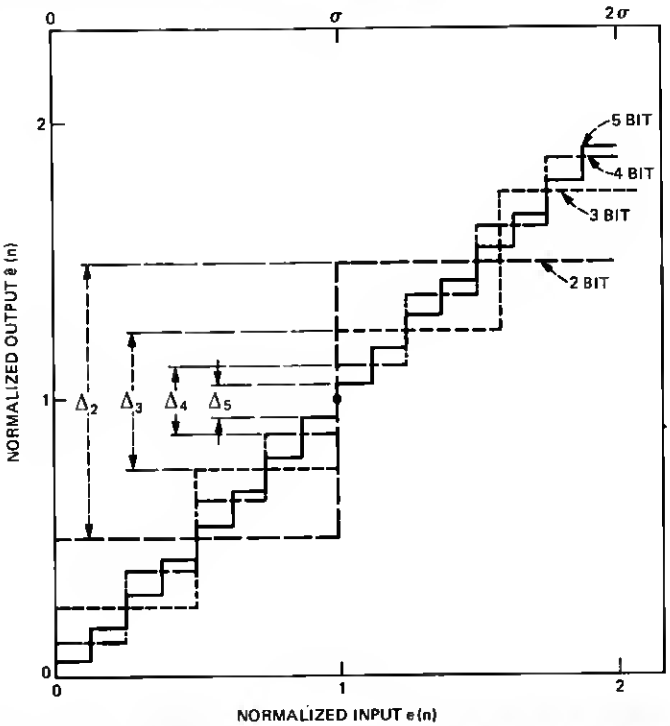

Fig. 11—Quantizer characteristic for 2- to 5-bit/sample characteristics.

The M values that were used here resulted in approximately a $\pm 2\sigma$ loading factor for each of the quantizer characteristics which is close to the optimum loading (in the mean-square error sense) proposed by Max.[13] This results in a quantizer characteristic centered about the variance of the signal as illustrated in the scale above Fig. 11. The estimate $\sigma(n)$ is therefore identified as the center of the quantizer (magnitude) characteristic which varies adaptively with the step-size adaptation.

4.4 Bit rate assignment

The buffer control determines the bits/sample assignment of the ADPCM coder, and it is based on the rate distortion relation in eq. (14). To give added flexibility to the algorithm, we also allowed a scale factor L in this equation to regulate the sensitivity of the bit allocation decision. This relation has the form

$$R(n) = L \log_2\left(\frac{\sigma^2(n)}{d_s^2(n)}\right), \tag{20}$$

where L is a parameter that can be adjusted.

4.4.1 Open loop control

To obtain an understanding of the range of values that L and $d_s^2(n)$ can take, we first ran the variable rate coder with an unlimited size input buffer and an open loop control of the bit assignment (i.e., $d_s^2(n)$ was fixed). The parameters L and $d_s^2(n)$ were adjusted as control parameters and the bit allocation was chosen on the basis of eq. (20) rounded to the nearest value 2, 3, 4, 5, or 6. The average bit rate used to encode a single sentence was then measured as a function of L and $d_s^2(n)$.

Figure 12 shows a plot of the average bits/sample, $\bar{R}(n)$, used by the coder, for this sentence, as a function of L and $d_s^2(n)$. As seen in the plot, as L increases, $\bar{R}(n)$ becomes more sensitive to variations in $d_s^2(n)$. Also, the range of $d_s^2(n)$ over which the average coder bit rate is between 2 and 6 bits/sample is clearly observed in this figure. These results were found to be useful in establishing practical limits for $d_s^2(n)$ when the adaptive buffer control is used.

4.4.2 Closed-loop dynamic buffer control

In the closed-loop buffer control, a limited size buffer was used, and a bit/sample allocation was made once per 60-bit packet, as described earlier. The bit allocation $R_c(n)$ was made on the basis of the scaled-rate distortion relation of eq. (20), and the allowed distortion $d_s^2(n)$ was "slowly" varied according to the relation in eq. (17). A two-piece approximation to the multiplier value $H(b(n-1)$ (see Fig. 8) was used in simulations according to the relation

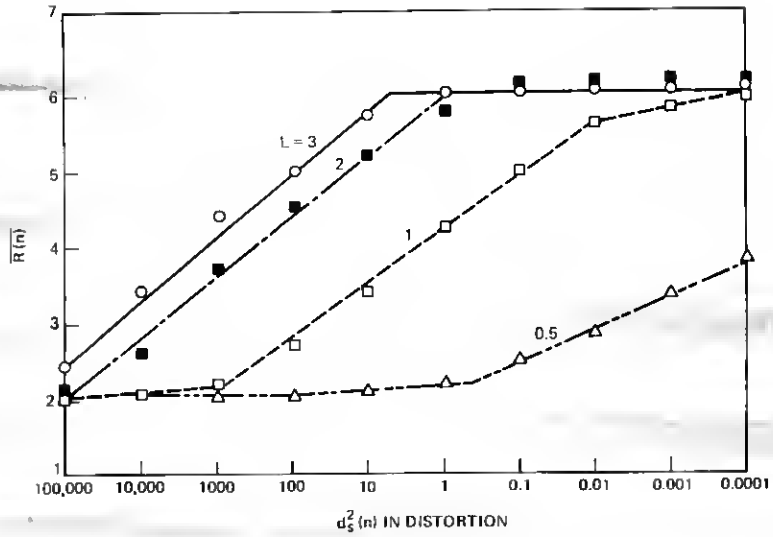

Fig. 12—Range of $\bar{R}(n)$ as a function of L and $d_s^2(n)$ for an open loop control.

$$H(b(n-1)) = \begin{cases} A > 1, & \text{if } b(n-1) \geq N/2 \\ 1/A < 1, & \text{if } b(n-1) < N/2, \end{cases} \tag{21}$$

where $b(n-1)$ is the number of samples in the input buffer in Fig. 9 and N is the size of the buffer. The value of A is greater than 1 and can be adjusted to control the speed at which $d_s^2(n)$ is allowed to vary. In general, the algorithm attempts to keep the buffer approximately one-half full (on the average).

If the buffer becomes full or empty, an additional constraint on the number of bits/sample, equivalent to that of eq. (16), is imposed to keep the buffer from overflowing or underflowing.

4.5 Performance of the variable rate ADPCM coder

The operation of the variable rate coder was observed with various parameters. In this section, we briefly illustrate the effects of some of these parameters.

Figure 13 shows a typical response of the variable rate coder for the sentence, "A lathe is a big tool." The parameters of the coder were:

N	= buffer size	= 1024
L	= rate distortion scale factor	= 1
A	= buffer adaption parameter	= 1.05
R_c	= fixed channel bit rate	= 32 kb/s
S_i	= input sampling rate	= 8 kHz

Figure 13a shows the input speech waveform, Fig. 13b shows the

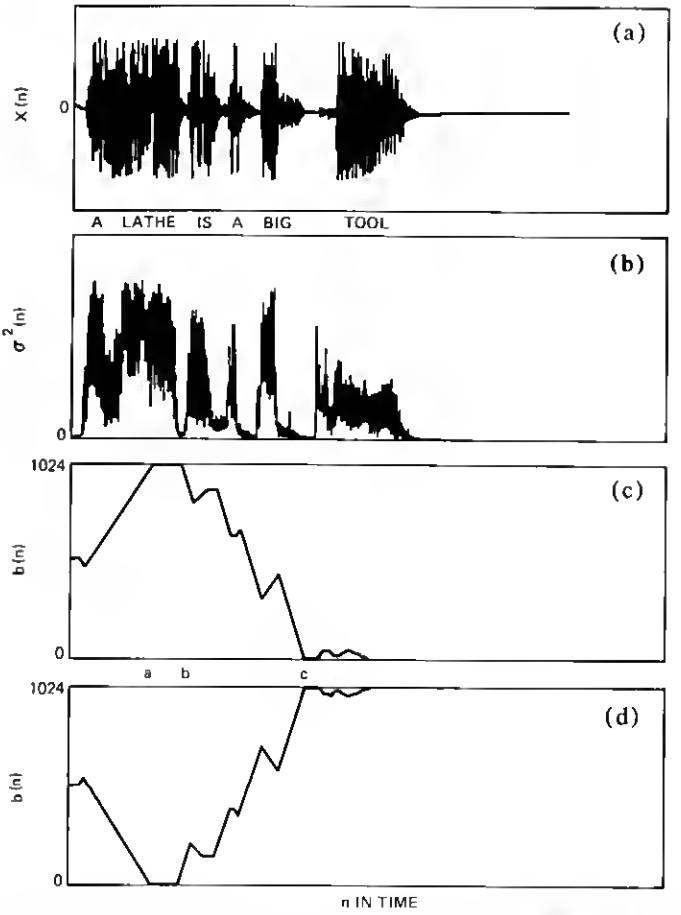

Fig. 13—(a) Input speech waveform. (b) Variance $\sigma^2(n)$. (c) Transmitter buffer status. (d) Receiver buffer status for the variable-rate ADPCM coder.

variance estimate of the difference signal e (n), and Figs. 13c and 13d show the number of samples in the transmitter and receiver buffers respectively. It can be noted that the receiver buffer status is the complement of the transmitter buffer status as discussed in Section 3.2.

As seen in Fig. 13, when the signal maintains a high level of activity, the transmitter buffer fills to capacity. When it becomes full, at time a (see Fig. 13c), the bit rate of the coder is limited to that of the channel rate to prevent overflow. At time b, the speech activity drops and the buffer begins to drain out. It fluctuates with speech activity until a silent region is encountered at time c. At this point, the coder rate is again fixed to that of the channel rate to prevent underflow of the buffer.

The effects of buffer adaptation corresponding to values of $A = 0.99$, 1.0, 1.025, 1.05, 1.11, and 1.2 are shown in Fig. 14b for the same sentence with a buffer size of $N = 1024$ samples. It can be seen that, when A is less than one, the buffer control is unstable and as A becomes larger ($\approx$1.2) the activity of the buffer is reduced. Figure 14c shows a similar result for a buffer size of $N = 256$ samples and it can be seen that, for smaller buffer sizes, the buffer fills up or drains out more often.

4.6 Recovery from channel errors

As pointed out earlier, the synchronization of the transmitter and receiver in the algorithm of Fig. 9 is unaffected by channel errors in the data. Resistance to these types of errors can be improved with a robust modification of the ADPCM step-size adaption algorithm.[14]

An error in the header, however, can result in an incorrect bit allocation in the receiver and the loss of a 60-bit packet of data. In addition, the receiver buffer will receive an incorrect number of samples resulting in an audible click and a misalignment of the transmitter

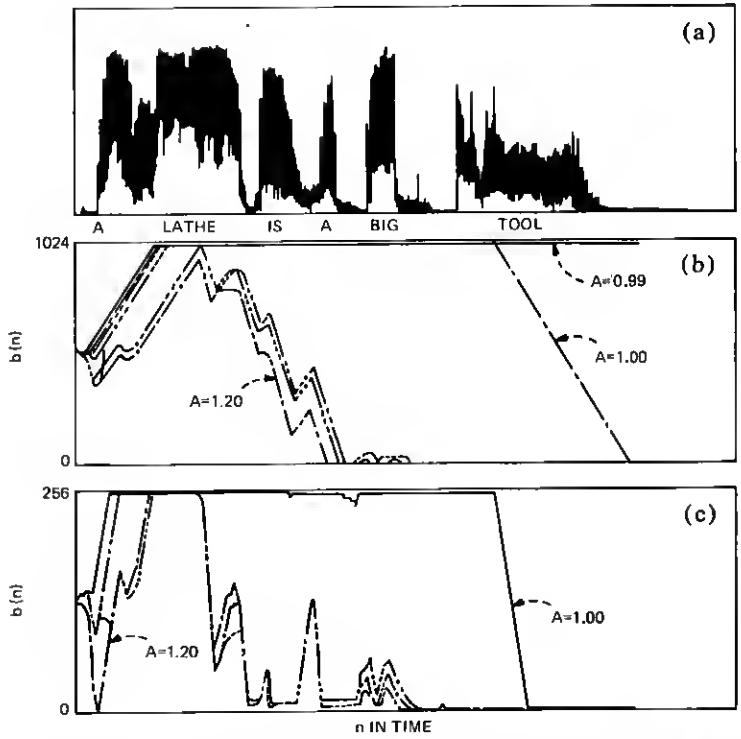

Fig. 14—(a) Variance of input speech waveform. (b) Buffer status for $A = 0.99$, 1.0, 1.025, 1.05, 1.1, and 1.2 (block size = 1024 samples). (c) Buffer status for $A = 0.99$, 1.0, 1.025, 1.05, 1.1, and 1.2 (block size = 256 samples).

and receiver buffers. This misalignment results in a temporary time shift between the transmitter and receiver (i.e., a change in total input to output delay) which is not audible to the listener.

A re-alignment of the receiver buffer will occur automatically when the buffer becomes full or empty, at which time a second signal error and time shift will occur. Two types of errors can occur, depending on whether the receiver buffer has an excess of samples or is missing samples. The first type of error is corrected when the receiver buffer becomes full and the second type of error is corrected when the receiver buffer becomes empty. In the first case, if the receiver buffer has more samples than it should and is driven into overflow (the transmitter buffer becomes empty), the excess samples can simply be discarded and the receiver and transmitter are again realigned. Since this occurs during a condition of low speech activity or silence, the loss of samples during this time is generally not audible. In the second case, when the receiver buffer is missing samples, the buffer will become empty prematurely during a period of high-speech activity (when the transmitter buffer fills up). In this case, zero-valued "dummy" samples can be inserted until realignment occurs between the transmitter and receiver. This silent period inserted during an active speech interval is not usually detectible by a listener. As a result, the realignment phases following an overflow or underflow error condition do not (in general) disrupt the audible speech.

Figure 15 is an example of a simulation of error recovery. Figure 15a shows the speech waveform and Fig. 15b shows the receiver buffer status. At time a (Fig. 15b), a header error was encountered resulting in an excess number of bits in the buffer, indicated by the shaded regions. At time b, during low speech activity, re-alignment with the transmitter buffer occurs. Following the error at a, no effects of the misalignment and realignment were audible to the listener.

V. ADDITIONAL CONSIDERATIONS AND COMMENTS ON VARIABLE RATE CODING

In this section, we examine a number of additional issues concerned with variable rate coding and comment on further directions and potential applications that need to be investigated.

5.1 Interaction of prediction gain and rate distortion criteria

In the coder example in Section IV (and the theoretical analysis in Section III), we have used the variance of the difference signal $e(n)$ (see Fig. 10) in controlling the buffer feedback. In this section, we briefly show the relationship between this variance and the signal variance of the input signal $x(n)$ for the case of a first-order predictor and demonstrate how the prediction gain interacts with the buffer

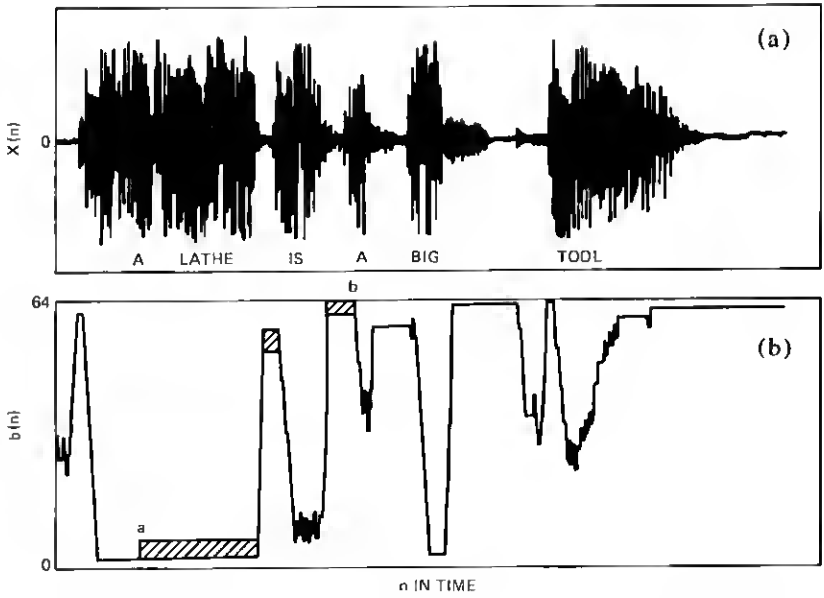

Fig. 15—Recovery of the buffer alignment after channel errors. (a) Speech waveform. (b) Receiver buffer status.

control. The interaction between the first-order predictor gain and the buffer control is also examined.

The differential signal $e(n)$ is (see Fig. 10)

$$e(n) = x(n) - \alpha\hat{x}(n). \tag{22}$$

After substituting the first-order correlation,

$$c = \frac{<x(n)\hat{x}(n-1)>}{<x(n)>}, \tag{23}$$

the expected value of $e^2(n)$ becomes

$$<e^2(n)> = <x^2(n)> \cdot [1 - 2\alpha c + \alpha^2]. \tag{24}$$

The result is that the difference signal variance is equal to the input signal variance multiplied by a factor dependent upon the signal correlation. Typically, for voiced speech, c corresponds to a signal correlation on the order of 0.9 (depending on the sampling rate) and a typical value of α might be about 0.9 for a fixed predictor. The result is that, for voiced speech the variance $<e^2(n)>$ is approximately proportional to the input variance, i.e.,

$$<e^2(n)>\Big|_{\substack{\text{voiced}\\ \text{speech}}} \approx 0.2 < x^2(n)>. \tag{25}$$

During unvoiced speech, the signal correlation c is on the order of 0.1 and

$$\langle e^2(n)\rangle \Big|_{\substack{\text{unvoiced}\\ \text{speech}}} \approx 1.6 \langle x^2(n)\rangle. \tag{26}$$

A comparison of the input signal variance $\langle x^2(n)\rangle$ and the difference signal variance $\langle e^2(n)\rangle$ is shown in Figs. 16b and 16c, respectively. Both variances appear to track relatively closely during voiced regions. During the unvoiced sounds, as for example /t/ in the word "tool," this loss of prediction increases the variance $\langle e^2(n)\rangle$. Since the bit allocation is based on $\langle e^2(n)\rangle \cong \sigma^2(n)$, this implies that a larger number of bits will be used to encode these unvoiced regions where the prediction gain becomes low but where the input signal variance is still significant.

5.2 Alternative criteria for buffer control based on code word magnitude

Throughout this paper, we have assumed that the buffer control is driven by the signal variance $\sigma^2(n)$ which is a result of the application of rate distortion theory. From the point of view of speech quality and perception, however, it is not clear that signal variance is the most appropriate parameter to be used for driving the bit allocation and buffer control.[9,10] Other, more perceptually meaningful parameters might be used as a driving function to produce better performance for speech.

In this section, we allude to one alternative candidate for this driving function based on a short-time average of the "code word energy." This function has been shown to be a sensitive indicator of speech and nonspeech activity.[15,16] The short-time, code-word energy is defined as

$$E(n) = \sum_{m=n-J}^{n} q_B \cdot |c(m)|, \tag{27}$$

where J is the number of samples over which the code word energy is averaged, $c(m)$ is the code word at time m (see Fig. 10), and q_B is a scale factor which normalizes the code words for different numbers of bits/sample. The code word $c(m)$ is more specifically defined as the quantizer level (see Fig. 11) expressed as an integer. The presence of small-magnitude code words are associated with silence, and the presence of large-magnitude code words are associated with speech.

Figure 16d illustrates an example of the short-time code word energy, $E(n)$, for parameters $J = 80$, and $q_2 = q_3 = q_4 = q_5 = 1$. It can be seen that $E(n)$ provides a more sensitive indication of speech activity than $\langle x^2(n)\rangle$ or $\langle e^2(n)\rangle$ and therefore may be a perceptually more desirable driving function to use for bit allocation and buffer control. It also provides a more reliable indication of when silence occurs in the

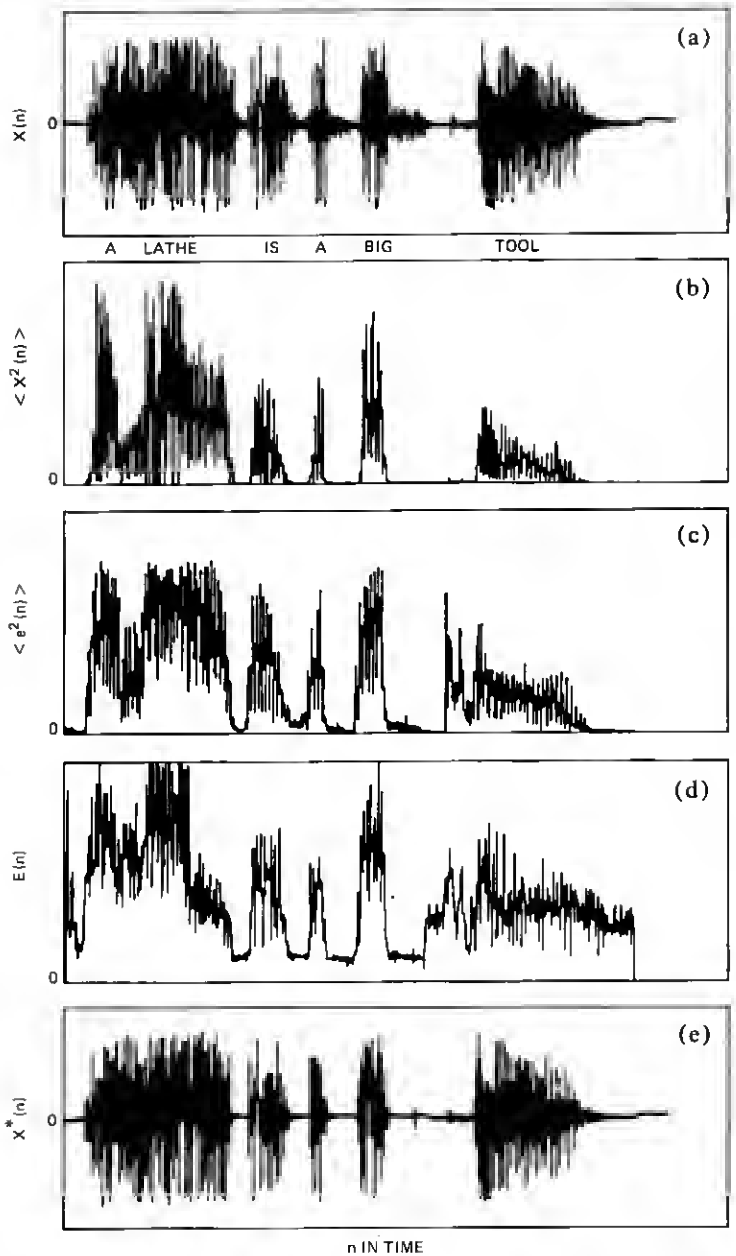

Fig. 16—(a) Input speech waveform. (b) input signal variance $<x^2(n)>$. (c) Difference signal variance $<e^2(n)>$. (d) Short-time code word energy $E(n)$. (e) Speech waveform with silence decision based on code word energy.

sentence.[15,16] For example, Fig. 16e shows the speech waveform $x(n)$ with the silence regions set to zero. It was not possible to distinguish between this sentence and the original in Fig. 16a when listening. This silence detector feature may be useful, particularly for a multiple-user application where some users can be turned off during silence.

5.3 Multiple user (TASI) applications of variable rate coding

In Section II, we illustrated a block approach for implementing variable-rate coding with multiple users and have pointed out that TASI-type advantages can be gained with this approach. A similar approach can also be implemented using dynamic buffering for each speaker. This idea is intuitively appealing in the sense that it couples TASI advantages with variable-rate coding advantages, i.e., it is a TASI with memory. By buffering the inputs of the speakers, bursts of strong activity from some speakers can be time-aligned with micro-silence regions of other speakers. Speakers whose buffers are full can receive short-time priority over other speakers whose buffers are not full. Thus, the statistics of speech activity seen by the channel is a combination of activity over time as well as across speakers. Flexible tradeoffs should be possible between the size of the input buffers (i.e., time delay) and the number of allowed users in the system.

VI. CONCLUSIONS

In summary, we can draw a number of conclusions concerning variable-rate coding:

(*i*) A block processing analysis shows that, for a single user, the improvements in block s/n of a variable-rate coder over that of a fixed-rate coder are dependent on the nonstationarity of the source and are related to the ratio of the arithmetic-to-geometric means of the signal variance.

(*ii*) For a single speech source, block sizes greater than about 100 ms are required before any substantial improvement over fixed-rate coding can be realized. Alternatively, flexibility in transmission rate is obtainable with very short block sizes with no loss in performance over fixed rate coding.

(*iii*) A multiple user variable-rate coding offers an interesting approach to implementing a TASI system.

(*iv*) Practical methods exist for designing variable-rate coders, and they can be made to be robust to channel errors.

REFERENCES

1. K. Bullington and J. M. Fraser, "Engineering Aspects of TASI," B.S.T.J., *38*, No. 2 (March 1959), pp. 353–364.
2. J. M. Fraser, D. B. Bullock, and N. G. Long, "Over-all Characteristics of a TASI System," B.S.T.J., *41*, No. 4 (July 1962), pp. 1439–1454.

3. J. P. Adoul and F. Daaboul, "Digital TASI Generalization with Voiced/Unvoiced Discrimination for Tripling T1 Carrier Capacity," Proc. IEEE Int. Conf. Commun., Chicago, June, 1977, pp. 32.5–310 to 32.6–314.
4. B. Gold, "Digital Speech Networks," Proc. IEEE, *65,* No. 12 (December 1977), pp. 1636–1658.
5. S. A. Webber, C. J. Harris, and J. L. Flanagan, "Use of Variable-Quality Coding and Time-Interval Modification in Packet Transmission of Speech," B.S.T.J., *56,* No. 8 (October 1977), pp. 1569–1573.
6. T. Berger, *Rate Distortion Theory—A Mathematical Basis for Data Compression,* Englewood Cliffs, N.J.: Prentice-Hall, 1971.
7. J. Huang and P. Schultheiss, "Block Quantization of Correlated Gaussian Random Variables," IEEE Trans. Commun. Sys., *CS-11* (September 1963), pp. 289–296.
8. R. Zelinsky and P. Noll, "Adaptive Transform Coding of Speech Signals," IEEE Trans. Acoust., Speech, and Sig. Proc., *ASSP-25* (August 1977), pp. 299–309.
9. R. E. Crochiere, L. R. Rabiner, N. S. Jayant, and J. M. Tribolet, "A Study of Objective Measures for Speech Waveform Coders," Proc. of the 1978 Zurich Seminar on Digital Communications, Zurich, Switzerland, March 1978.
10. J. M. Tribolet, P. Noll, B. J. McDermott, and R. E. Crochiere, "Complexity vs. Quality for Speech Waveform Coders," Proc. of the IEEE Int. Conf. on Acoust., Speech, and Sig. Proc., Tulsa, Okl, April 1978, pp. 586–590.
11. N. S. Jayant, "Digital Coding of Speech Waveforms: PCM, DPCM, and DM Quantizers," Proc. IEEE, 62, (May 1974), pp. 611–632.
12. A. G. Tescher and R. V. Cox, "Image Coding: Variable Rate DPCM Through Fixed Rate Channel," Proc. Society of Photo-Optical Instrumentation Engineers, *119* (August 1977), pp. 147–154.
13. J. Max, "Quantizing for Minimum Distortion," IRE Trans. Inform. Theory, *IT-16* (May 1960), pp. 7–12.
14. D. J. Goodman and R. M. Wilkinson, "A Robust Adaptive Quantizer," IEEE Trans. Comm., November 1975, pp. 1362–1365.
15. R. W. Schafer, J. J. Dubnowski, K. Jackson, and L. R. Rabiner, "Detecting the Presence of Speech Using ADPCM Coding," IEEE Tran. Comm., May 1976, pp. 563–567.
16. L. H. Rosenthal, R. W. Schafer, and L. R. Rabiner, "An Algorithm for Locating the Beginning and End of an Utterance Using ADPCM Coded Speech," B.S.T.J., *53,* No. 6 (July-August 1974), pp. 1127–1135.